



Introduction to

ARDUINO 

How to use electronics to make your projects better!

Guest wifi:	HSNOTTS-guest	Password: hackspacebiscuits
Presentation:	http://wiki.nottinghack.co.uk/wiki/Arduino101	
Software:	http://arduino.cc/en/Main/Software	



Welcome!

- Let's get programming!
- What did we just do? What is an “Arduino” anyway?
- Serial communications (making your Arduino talk to your laptop)
- Electrical Basics: Voltage, current, resistance
- More blinking LEDs
- Using switches



Welcome!

- Lunch?
- Structure and Decisions: Order out of chaos
- Your Arduino is a wimp: giving it some muscle!
- Life isn't digital: reading real-world values
- Putting it all together

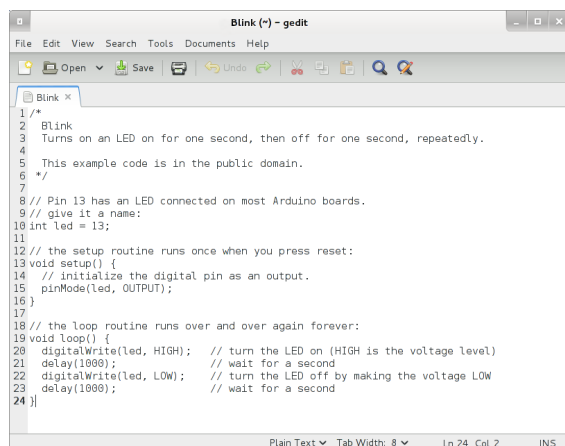


Dive in: Blink an LED

Code Along #1

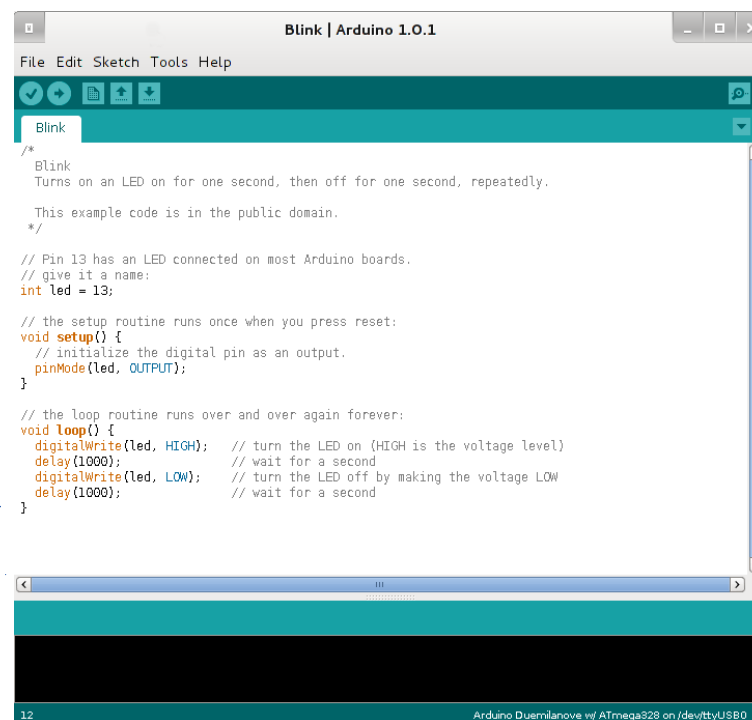


What just happened?



```
1 /*  
2  * Blink  
3  * Turns on an LED on for one second, then off for one second, repeatedly.  
4  *  
5  * This example code is in the public domain.  
6  */  
7  
8 // Pin 13 has an LED connected on most Arduino boards.  
9 // give it a name:  
10 int led = 13;  
11  
12 // the setup routine runs once when you press reset:  
13 void setup() {  
14   // initialize the digital pin as an output.  
15   pinMode(led, OUTPUT);  
16 }  
17  
18 // the loop routine runs over and over again forever:  
19 void loop() {  
20   digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
21   delay(1000); // wait for a second  
22   digitalWrite(led, LOW); // turn the LED off by making the voltage LOW  
23   delay(1000); // wait for a second  
24 }
```

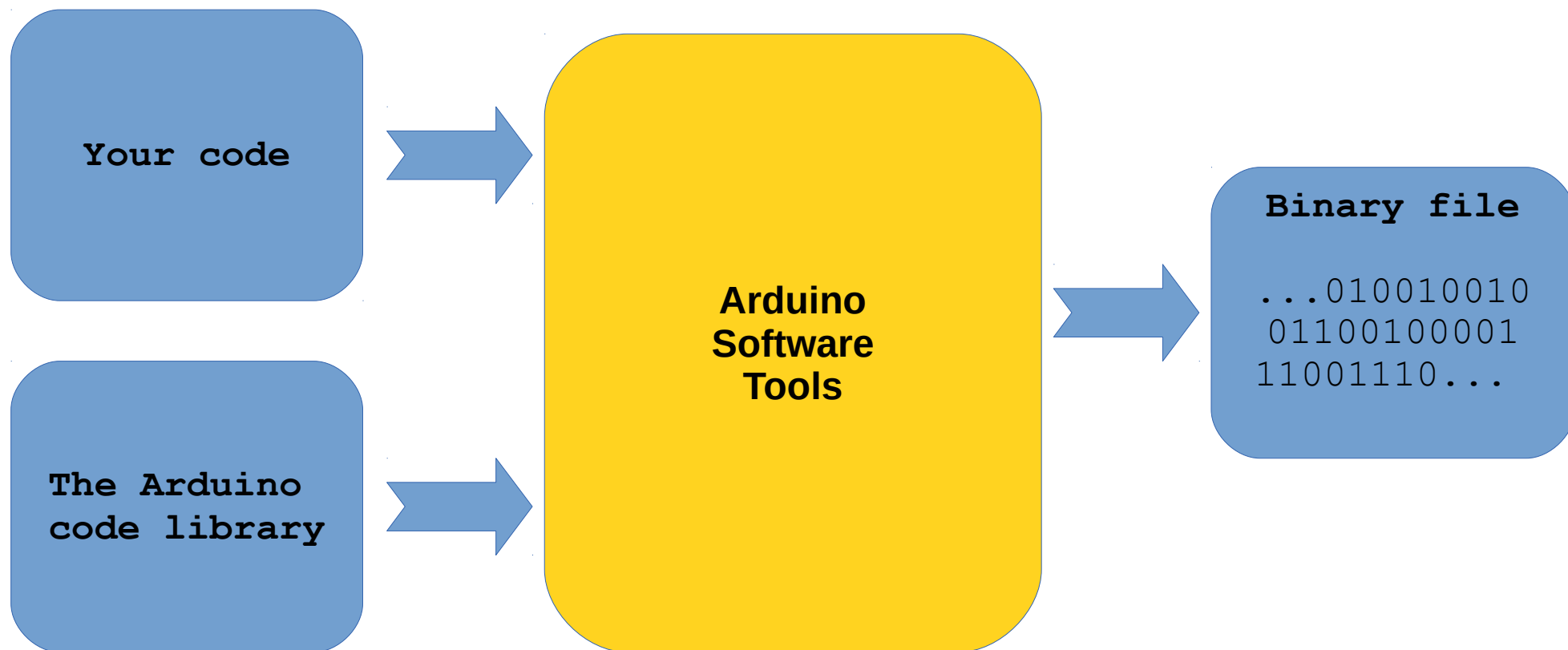
The Arduino IDE is a text editor combined with programming tools.



```
/*  
 * Blink  
 * Turns on an LED on for one second, then off for one second, repeatedly.  
 *  
 * This example code is in the public domain.  
 */  
  
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000); // wait for a second  
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW  
  delay(1000); // wait for a second  
}
```



What just happened?



When you press the “Upload” button, your code is combined with the Arduino library and made into a file that the Arduino microcontroller is programmed with.

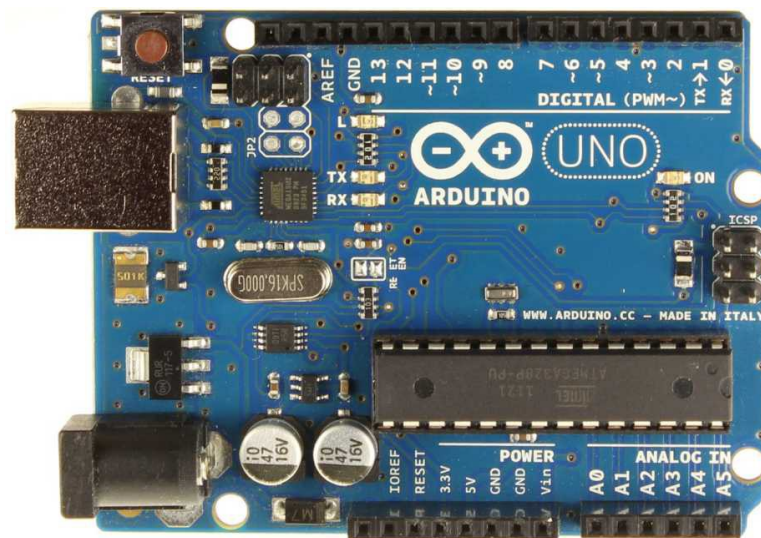


What just happened?

Binary file

```
...010010010  
01100100001  
11001110...
```

USB



... and then gets sent down the USB cable to the Arduino. Once there, it stays there until you re-program it, even if the power is removed.



Serial Monitor

Code Along #2



Variables

Code Along #3



Variables

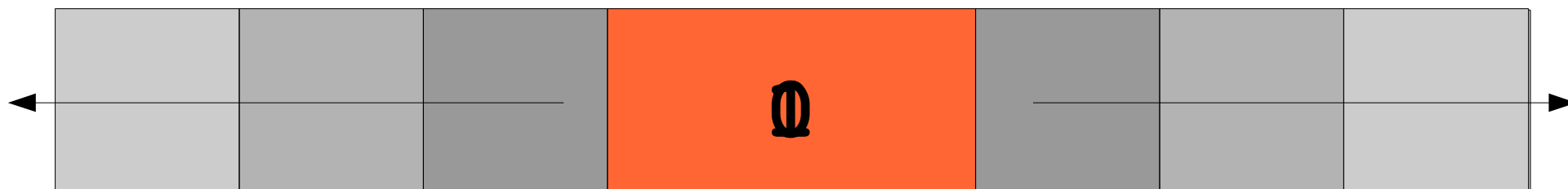
Named places to store your data



```
int loop_counter = 0;
```



Variables



```
loop_counter = loop_counter + 1;
```



Variables



```
loop_counter = loop_counter + 1;
```

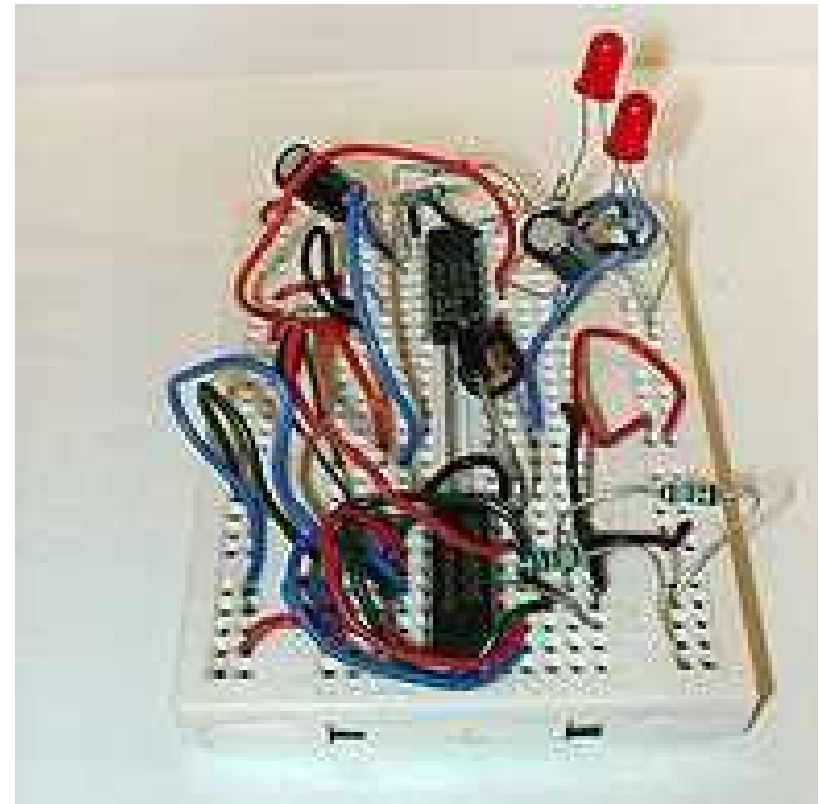
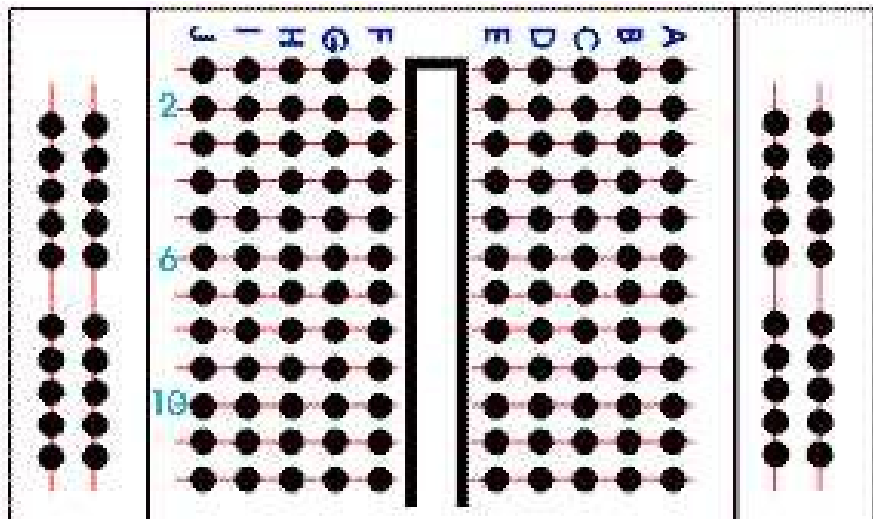


Variables: Size and Limits

<code>char</code>	Character 'a', 'b', 'c'... or -128 to 127	1 byte
<code>unsigned char</code>	0 to 255	1 byte
<code>int</code>	-32768 to 32767	2 bytes
<code>unsigned int</code>	0 to 65535	2 bytes
<code>bool</code>	true or false	1 bytes
<code>String</code>	String hello = "Hello!";	???



Breadboard





Using other components

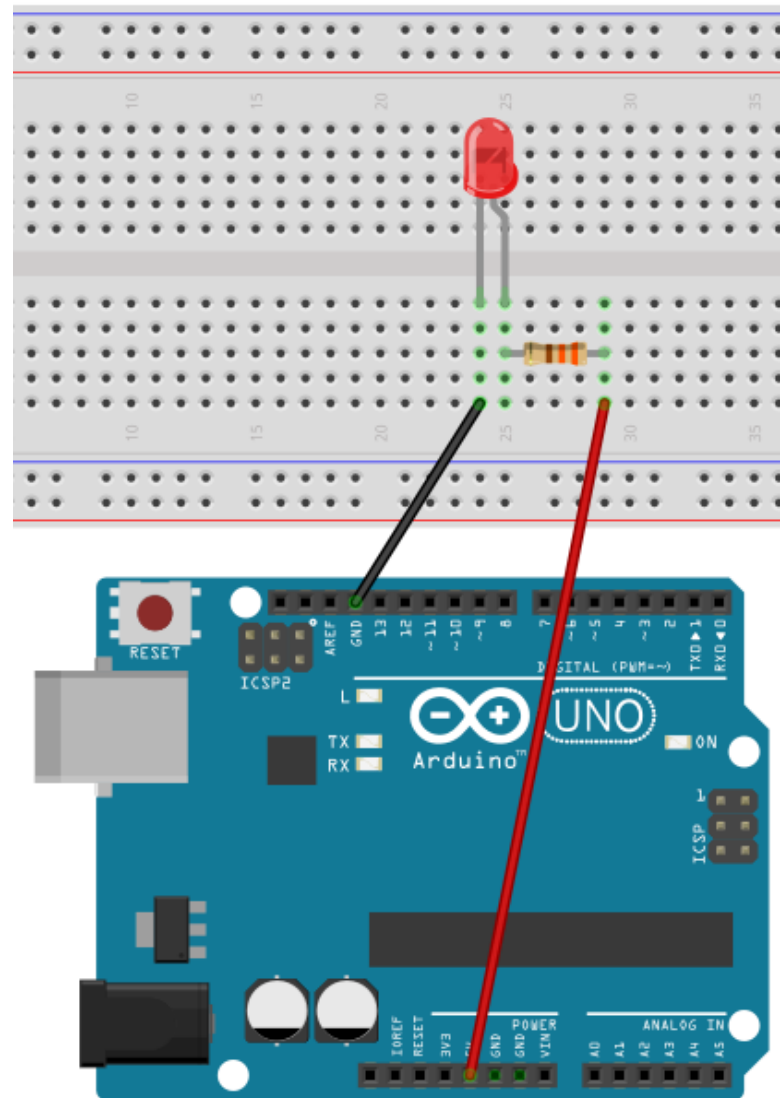
Construct this circuit using a 330R resistor.

Measure the voltage of the LED and resistor.

Calculate the current through the LED.

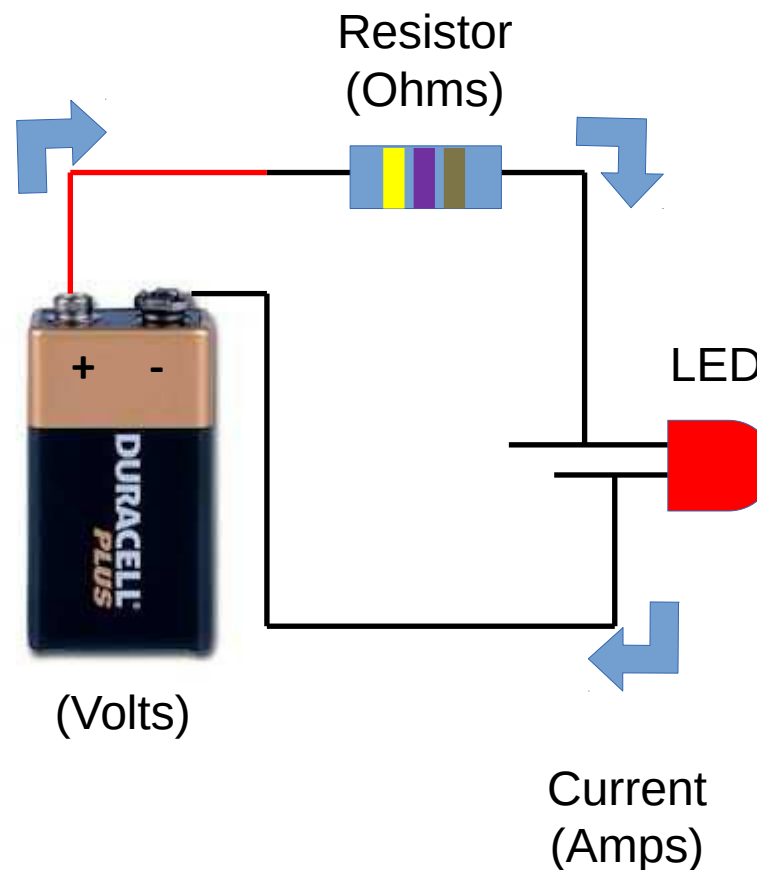
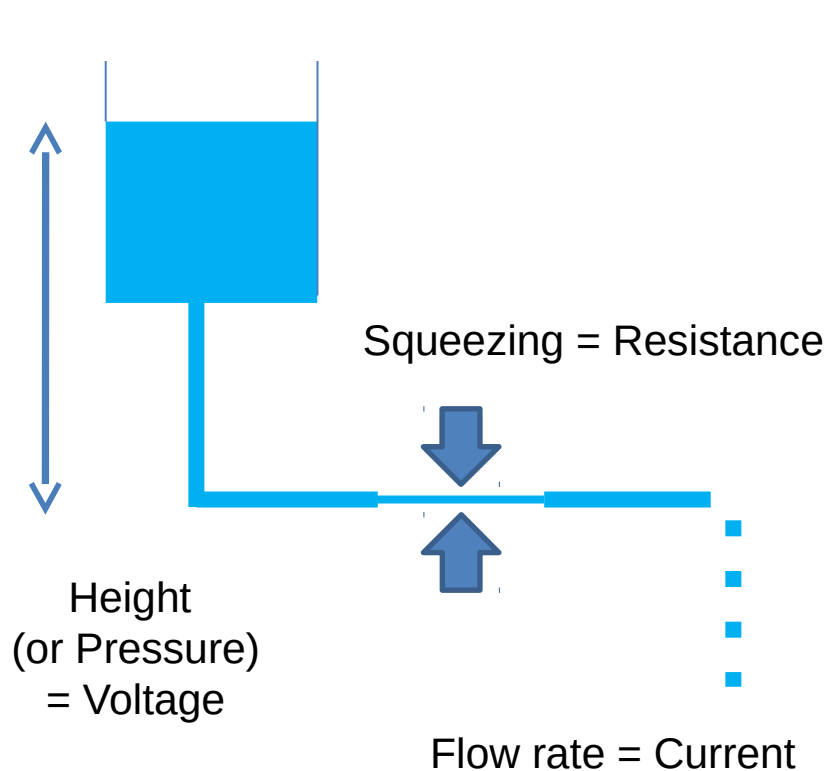
Replace the resistor with a 1K resistor. Does the voltage change? What is the new current?

Turn the LED round. Does it still light?





Voltage, Current, Resistance

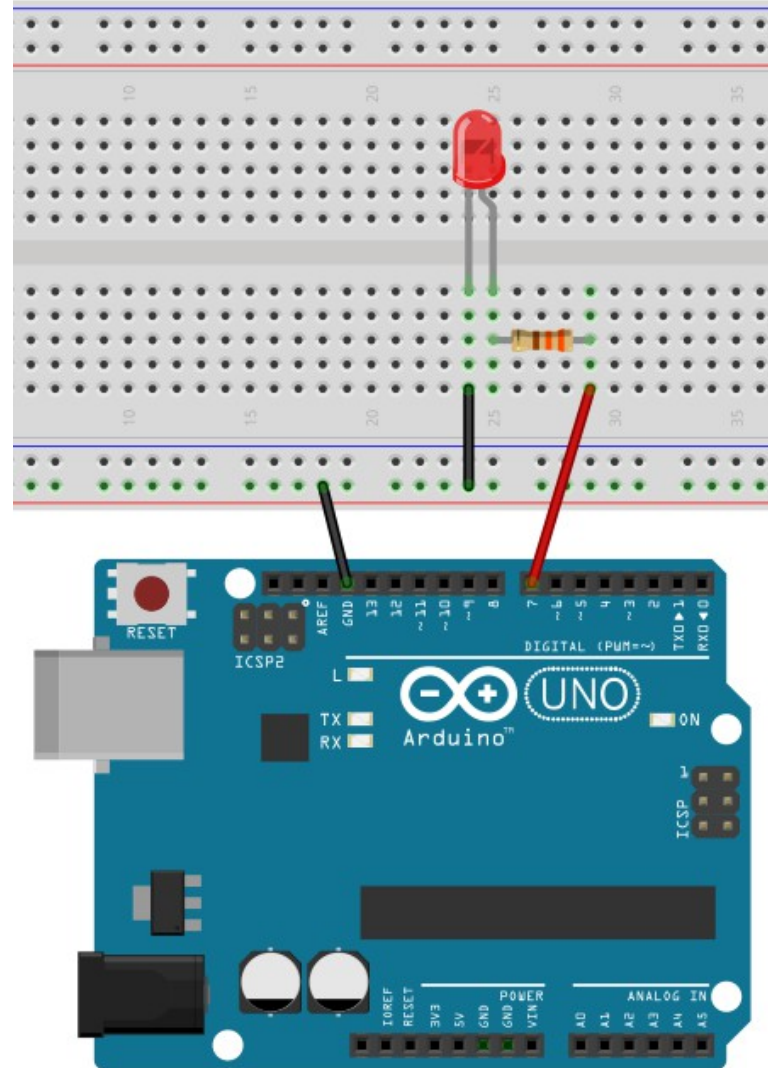




Another blinking LED

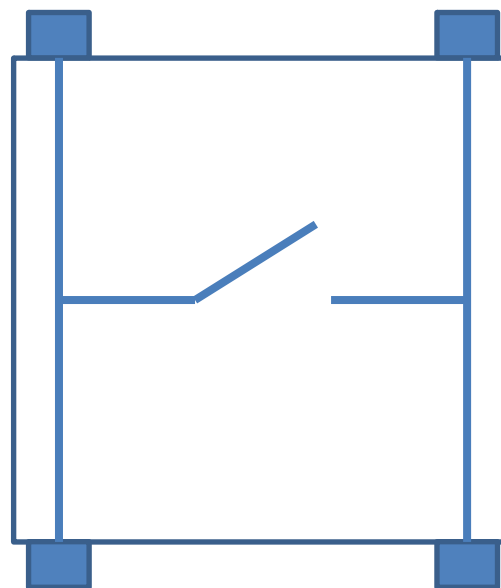
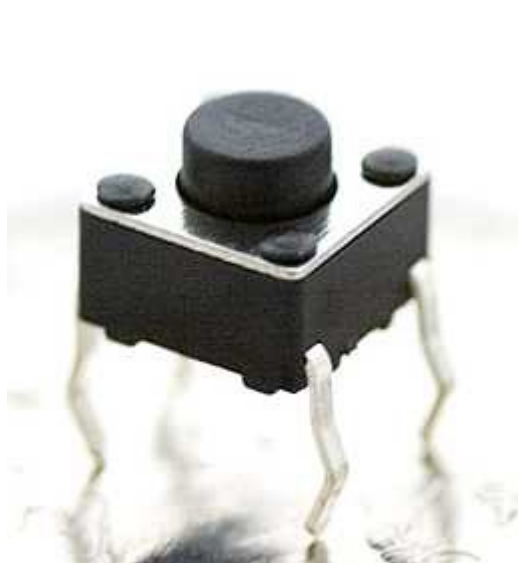
Change your breadboard circuit as shown.

Code Along #4





Tactile Switch

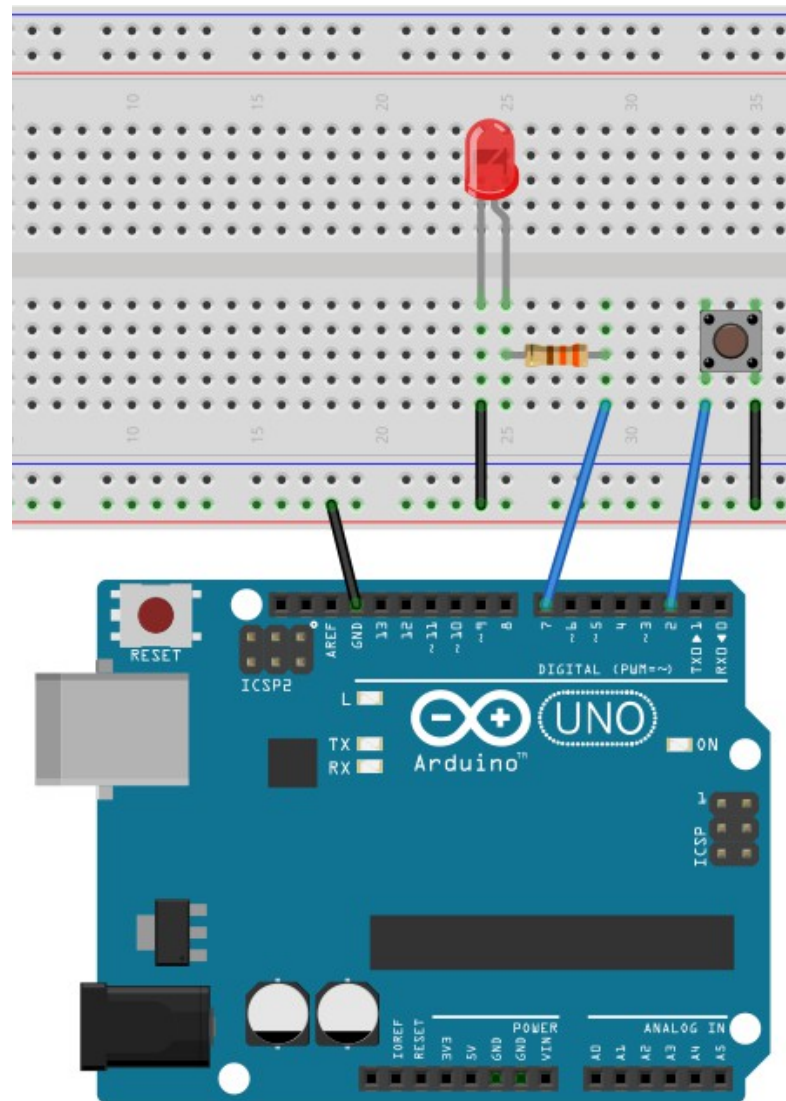




Switch controlling LED

Change your breadboard circuit as shown.

Code Along #5



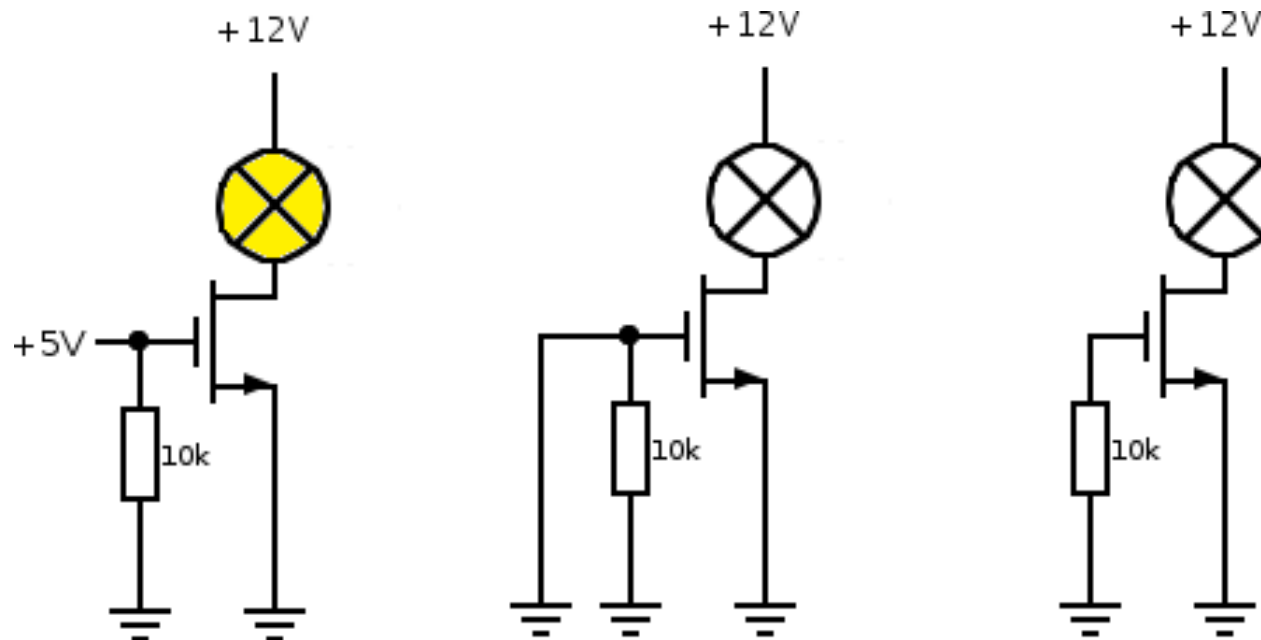
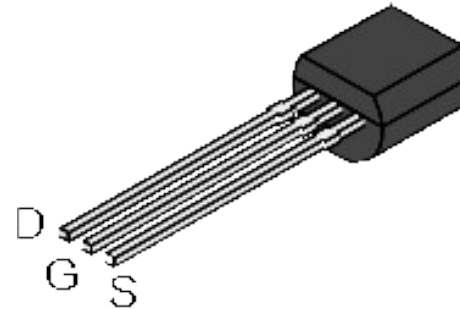
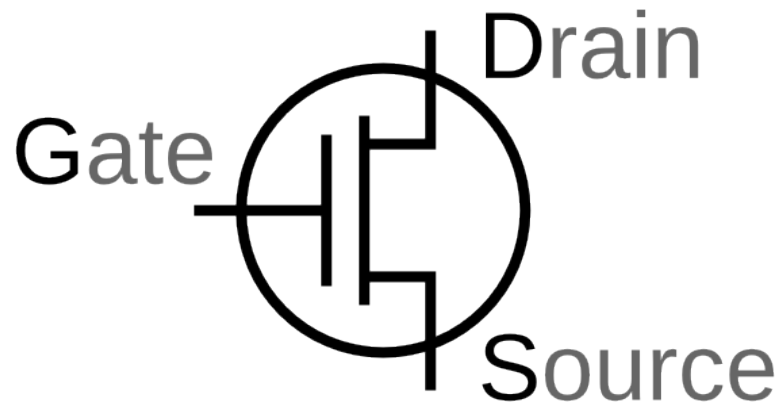


High Power Loads

- These devices need more current than the Arduino can supply.
- We use a transistor to increase the current available.
- Transistor acts as a Arduino controlled switch.
- Need protection from inductive loads (usually wound coils such as motors and relays).
- NPN Bipolar and N-channel MOSFET are commonly used with the Arduino.
- Voltage of the load (motors, etc.) can be higher than Arduino voltage.



N-Channel MOSFET

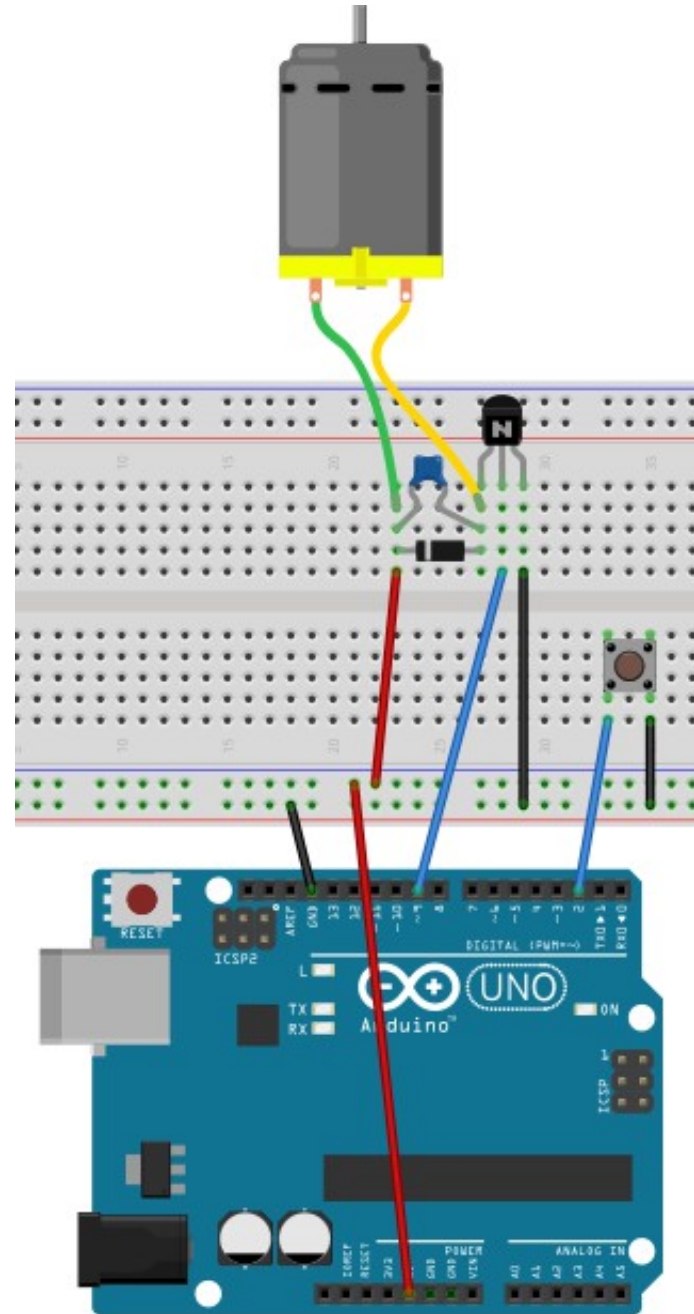




Motor Control

Change your breadboard circuit as shown.

Code Along #6





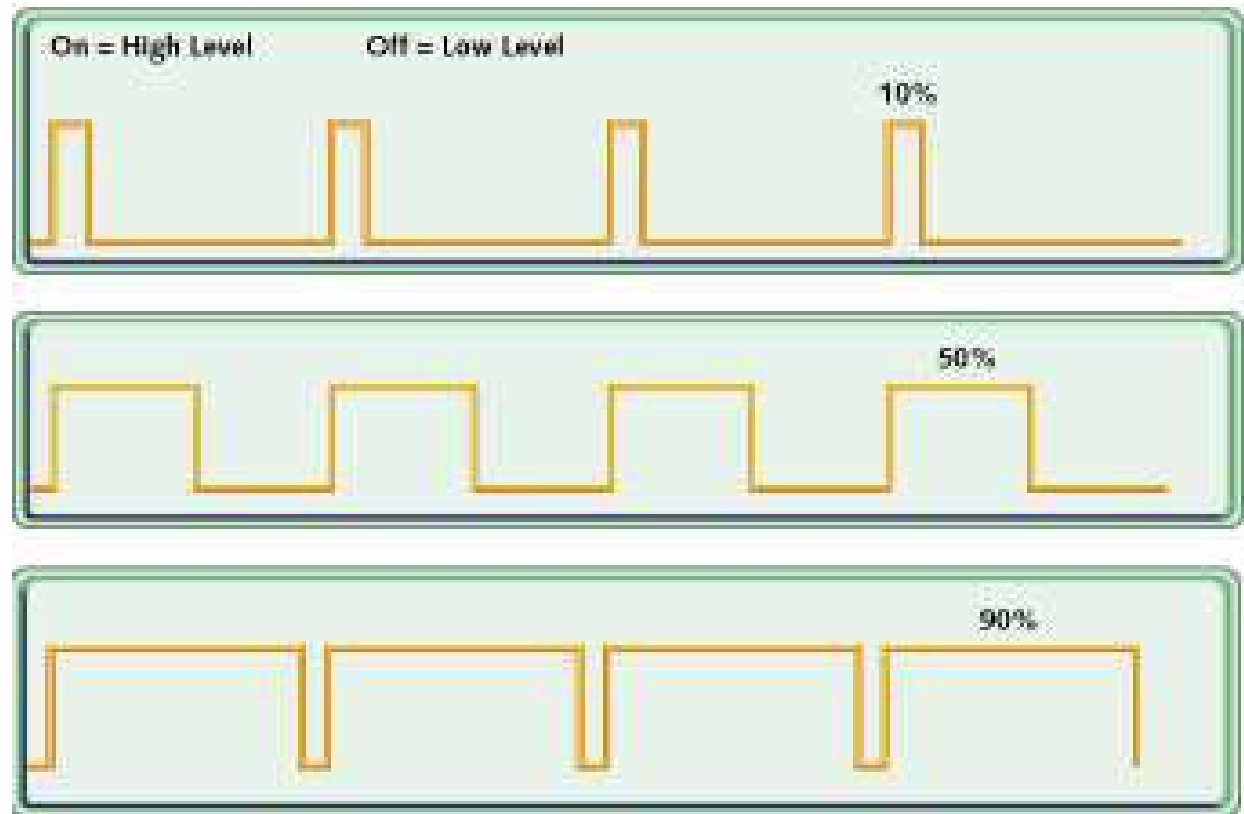
Pulse Width Modulation

Drives motor at full voltage.

This gives high torque even at low speeds.

Used on Arduino pins 3, 5, 6, 9, 10, 11

Value 0 - 255



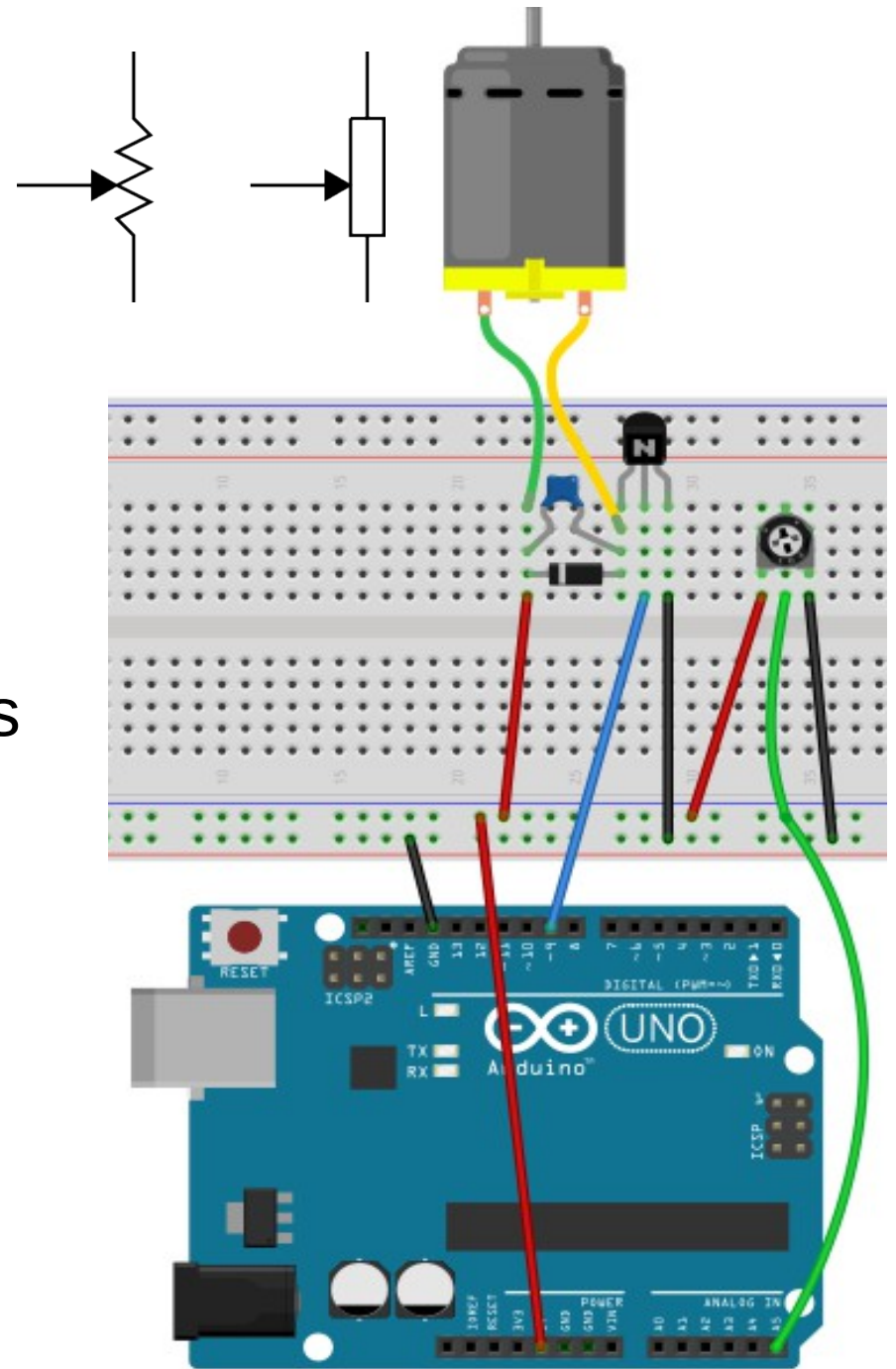


Reading real-world signals

- Arduino has 6 analog inputs
- Read with:
 - `int value = analogRead(pin);`
 - Pin is 0-5 or A0-A5
 - Value from 0 to 1023 representing voltage of 0-5V



Potentiometer (Variable Resistor)



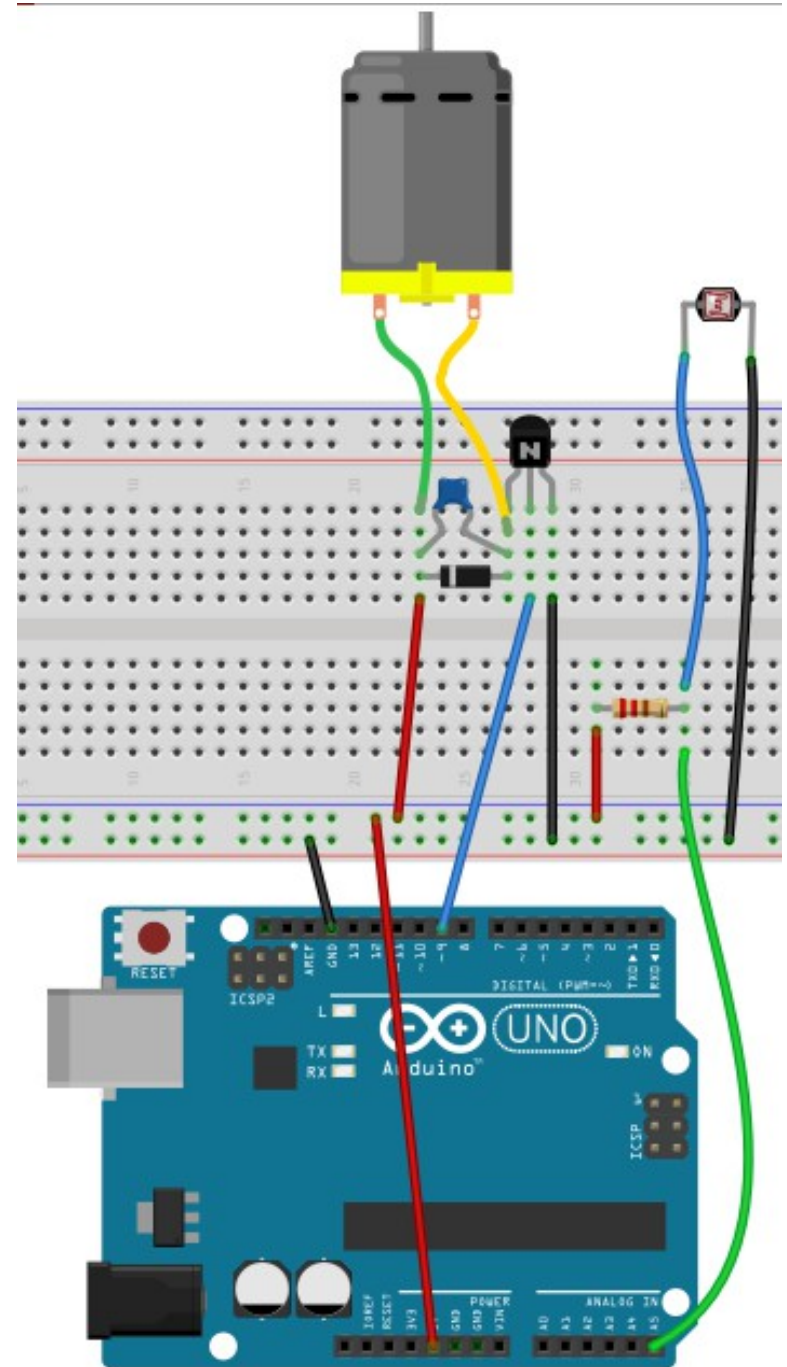
Change your breadboard circuit as shown.

Code Along #7



Light Dependant Resistor

Change your breadboard circuit as shown.





Doing more than one thing

Code Along #8

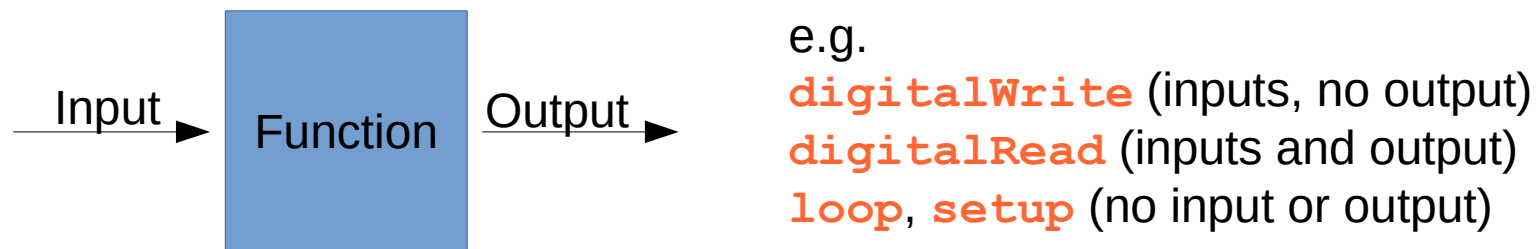


Functions

Named blocks of code that perform a particular job

- Arduino library provides *loads* of them for you.
- Using a function is known as “calling” or “invoking” it.
- **loop** and **setup** are functions in your sketch.
- You can write your own!

Functions may have *inputs* and *outputs*.





Functions

Code Along #9



That's It!